

AMENDMENTS TO THE SPECIFICATION

Please replace the following paragraphs.

Please replace paragraph [0001] with this paragraph:

--U.S. Patent Application Serial No. 09/899,473, entitled "Teamware Server Working over HTTP/HTTPS connections" and U.S. Patent Application Serial No. 09/900,763, entitled "Teamware Repository of Teamware Workspaces," both filed contemporaneously herewith, contain subject matter related to the disclosure herein.--

Please replace paragraph [0019] with this paragraph:

--Referring now to the accompanying drawings, Figure 1 shows a block diagram of a teamware system **5** that allows transactions to be executed between a local workspace **10** and a remote workspace **15**. The local workspace **10** is local in the sense that a teamware client **20** uses local access methods or a network file sharing protocol to access the contents of the local workspace **10**. The remote workspace **15** is remote in the sense that the teamware client **20** does not use local methods or a network file sharing protocol to access the contents of the remote workspace **15**. In the figure, the remote workspace **15** is shown inside a repository **25**, such as described in U.S. Patent Application Serial No. 09/900,763, entitled "Teamware Repository of Teamware Workspaces," herein incorporated by reference. In accordance with this embodiment, the teamware system **5** includes a teamware server **30** that manages and provides access to the repository **25** and remote workspace **15**. The teamware client **20** communicates with the teamware server **30** via an application programmer interface (API). The teamware client **20** executes the transaction logic locally and sends certain commands to the teamware server **30** to be executed at the teamware server **30**. The result, such as the content of a file, an object, or an exception, is returned to the teamware client **20**.--

Please replace paragraph [0020] with this paragraph:

--The teamware client **20** and teamware server **30** may be software components written in Java™ or another programming language. At runtime, as shown in Figure 2, the teamware server **30** includes one or more server objects **35** (only one is shown) that implement the API. To execute a command at the teamware server **30**, a client object **40** invokes a server method (defined in the API) by sending a message to the server object **35**. The server object **35** executes

the corresponding operation and returns the result to the client object 40. If the teamware client 20 and teamware server 30 run in different virtual machines, the server object 35 is considered to be remote to the client object 40. If the virtual machines are located on the same computer or on separate computers connected via a network file sharing protocol, a mechanism such as Remote Method Invocation (RMI) can be used to remotely invoke the methods of the server object 35. If the virtual machines are on separate computers that are not connected by a network file sharing system, a mechanism such as described in U.S. Patent Application Serial No. 09/899,473, entitled "Teamware Server Working Over HTTP/HTTPS Connections," herein incorporated by reference, can be used to invoke the methods of the server object 35.--

Please replace paragraph [0021] with this paragraph:

--Figure 3 shows how the mechanism described in U.S. Patent Application Serial No. 09/899,473, entitled "Teamware Server Working Over HTTP/HTTPS Connections," works. To invoke a method of the server object 35, the client object 40 makes a method call to a proxy object 45 which has the same interface as the server object 35. The proxy object 45 forwards the method call to a helper object 50. The helper object 50 analyzes the method call, marshals the parameters included in the method call, and converts the method call to a HTTP request. The helper object 50 passes the HTTP request to a connection object 55, which sets up a connection 60 with a web server 65 and sends the HTTP request to the web server 65 over the connection 60. The connection 60 is based on HTTP or HTTPS protocol. The web server 65 passes the HTTP request to a web container 67. A servlet 70 and the server object 35 are instantiated inside the web container 70. The servlet 70 parses the request and delegates processing of the request to the server object 35. The server object 35 processes the request and returns the results to the servlet 35. The results are passed to the servlet 35, web container 67, and then the web server 65. The web server 65 sends a HTTP response (including the results) to the client object 40.--